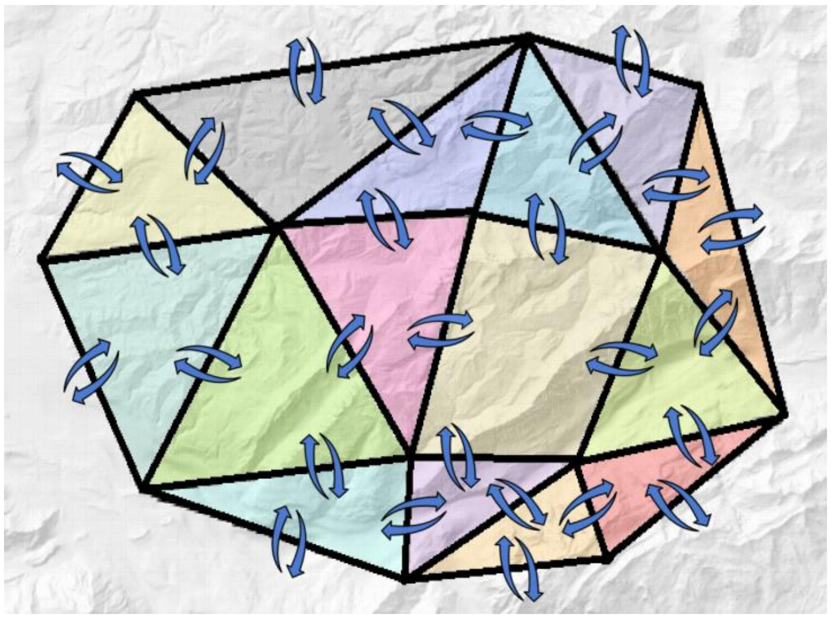
Seminar geodata analysis and modelling

CTS Root number: 438745

Institute of Geography, Faculty of Science, University of Bern

HYDROLOGICAL CONNECTIVITY

CREATING AN ARCGIS TOOLBOX



submitted by:	Florian Broghammer	13-927-322
	Julia Ryser	14-113-286

date of submission: 29.11.2019

supervised by: PD Dr. Andreas Paul Zischg

Dr. Pascal Horton

Table of contents

List of Figures and Tables

1. Summary

We created a toolbox “hydrological connectivity” which includes one script tool. This script tool needs to be saved in the same folder as the toolbox (ESRI, 2016a). The script was written and tested for the version ArcMap 10.6.

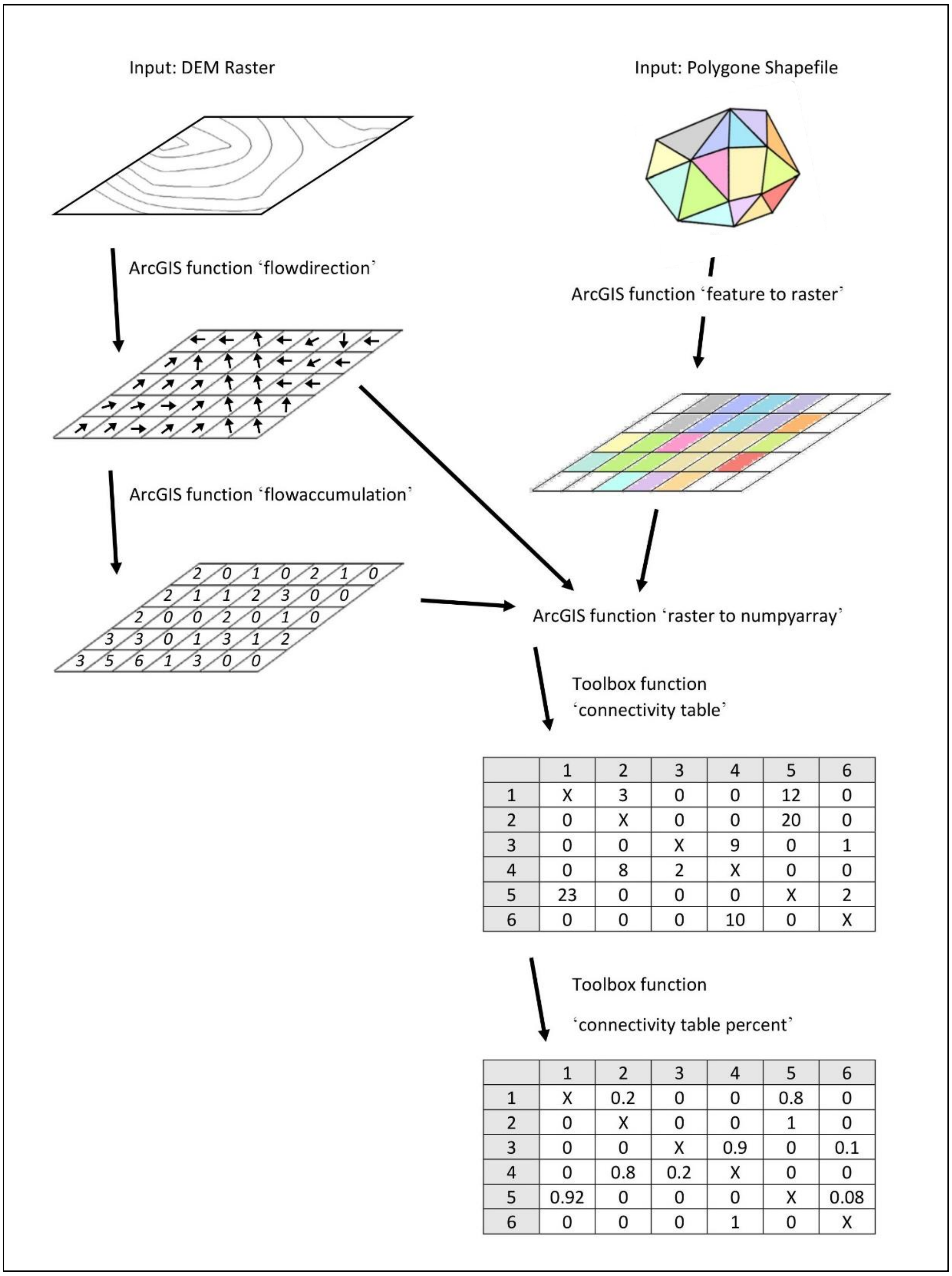
Starting with connected polygons of any shape, the code calculates how the outflow of a polygon is distributed to his neighbors. The script tool needs a digital elevation model (DEM) and the polygon shapes as input. It firstly processes the flow direction and flow accumulation. Based on these, two connectivity tables are then calculated. One table summarizes in percent and the other in total cell values, how much water flows from one polygon to the others. These tables are exported in two xls-files. The workflow is shown in Figure 1 on the next page.

The tables are to read in the way, that the number in the cell is the amount of water which flows from the polygon with the FID in the first column into the polygon with the FID in the first row. This means for the example connectivity table (Table 1), that 10 cells flow from polygon 1 (FID 1) into polygon 2 (FID 2) and 5 cells flow from polygon 2 into polygon 1.

Table 1: Connectivity table example.

FID	1	2
1	x	10
2	5	x

The toolbox can for example be used to calculate the hydrological connectivity between hydrologic response units (HRUs).


Input: DEM Raster
Input: Polygone Shapefile
ArcGIS function ‘flowdirection’
ArcGIS function ‘feature to raster’
ArcGIS function ‘flowaccumulation’
2 0 1 0 2 1 0
2 1 1 2 3 0 0
2 0 0 2 0 1 0
3 3 0 1 3 1 2
3 5 6 1 3 0 0
ArcGIS function ‘raster to numpyarray’
Toolbox function ‘connectivity table’
Toolbox function ‘connectivity table percent’

	1	2	3	4	5	6
1	X	3	0	0	12	0
2	0	X	0	0	20	0
3	0	0	X	9	0	1
4	0	8	2	X	0	0
5	23	0	0	0	X	2
6	0	0	0	10	0	X

	1	2	3	4	5	6
1	X	0.2	0	0	0.8	0
2	0	X	0	0	1	0
3	0	0	X	0.9	0	0.1
4	0	0.8	0.2	X	0	0
5	0.92	0	0	0	X	0.08
6	0	0	0	1	0	X

Figure 1: Workflow of the code.

2. Conditions and restrictions

The boarders of the polygon raster are set to the extent of the DEM. This means, if the polygon shape file has a bigger extend, it is cropped. If it's smaller or irregular shaped, the outside is treated as one polygon. To shorten the calculation time, it makes sense to use a DEM snipped which isn't much bigger than the polygon shapefile.

Cells, which flow out of the raster, are ignored. This means, that there's only limited information about the polygons at the edge of the raster available.

The polygons and the DEM need to be in the same projection. This is a potential error source, which wasn't further examined.

When polygons are not properly connected, the room in between is either seen as a polygon on its own, or if it's connected to the outside, calculated as the outside polygon.

3. Code structure and explanations

Header

The code starts with a header, which contains the most important information.

Imports

The needed modules are imported. Of special note here is the expression "from __future__ import division". This feature imports a Python 3 function into Python 2, allowing for integers to be divided by each other. This was vastly important for the results because some decimal results were rounded down to zero.

inputs, environment and workspace

The inputs from the user are set with the tool 'arcpy.GetParameterAsText'. The user needs to set a workspace folder, the DEM raster and a polygon file. By every new run, the old output is overwritten, if this is not desired, 'overwriteOutput' should be set to FALSE.
The input DEM raster will be used as a reference raster. All new calculated raster will have the same extend and cell size.

Definition of the functions connectivity_table and connectivity_table_percent

In this section, the two new functions are defined, but not yet summoned. The first function "connectivity_table ", hereafter called function 1, needs 4 inputs: The polygon shapefile that was read in earlier, a flow direction raster and a flow accumulation raster based on the used DEM as well as the workplace folder where these data are stored. The flow direction raster and flow accumulation raster will be calculated later by the script, not by function 1 itself although this would be possible as well. The reason this was not implemented was because during the development process of said function 1, on-demand access to an ArcGIS software was not available.

Function 1 starts by creating the aforementioned result table. In order to do this, an empty list called "ID_list" is created. This list receives a new entry every time a for-loop looping through every entry of the polygon shapefile detects an element which was not already in said list. The result will be a list of every ID value present in the polygon shapefile. The list is sorted at the end. The length of this list defines the dimension of the result table, it will have as many rows and columns as "ID_list" has entries.

Afterwards, function 1 creates a numpy zero array with the dimension NxN. N is the number of elements of "ID_list" increased by 1. Both the first columns and the first row are replaced with values of "ID_list". They serve as headers in order to read the table according to the scheme which was explained in the first chaper.

Next, a for loop is created to loop through all three raster datasets. Since they all have the same dimension, this could be done using only one loop. The flow direction is checked for every raster cell. The current row and column number is then increased or decreased according to the flow direction. This is done by a chain of "elif" statements.

Following up, a check is made if this destined cell shares the same ID as the current one. If not, the current cell would mark the border of a polygon. Therefore, the flow from the current cell to the destined cell would be a flow from the polygon with the ID of the current cell into the polygon with the ID of the destined cell.

If this is the case, the value of the flow accumulation of the current raster needs to be added to the result table. In order to do this at the correct location, a for loop is used to loop through the result tables first row and column. The loop will stop if the ID stored in the re-

sult table is the same as the ID of one of the two polygons. Both values are stored. They serve as the coordinates at which the addition needs to take place.
In other words, the for-loop determines that the ID of the current cell is located in the column number x. This marks the ID of the outflowing polygon. Another for-loop determines that the ID of the destined cell lies in row y. This is the ID of the polygon into which the flow accumulation of the current cell flows. This needs to be added to the already existing value in the result table at the coordinates x and y. Additionally, the cell itself also flows into the polygon, this is why the flow accumulation value is increased by one. In short: result table [x,y] = result table [x,y] + flow accumulation of the current cell + 1. If this process is done for every cell, the total amount of cells flowing from a polygon to another will be summed up. Once this process is finished, function 1 will export the result table as excel and store it in the same folder as the datasets were stored.

The second function "called connectivity_table_percent", hereafter called function 2, will create and export a result table similar to function 1. While function 1 uses an absolute value, the total number of cells flowing from a polygon into the other polygons, function 2 will display those results as percent. It does this by summarising all values of a row and dividing each entry in this row by this sum.

ArcGIS Comands

The sections execute GIS commands. First, the DEM gets filled to get rid of any depressions (ESRI, 2016c). These aren’t interesting for our approach. Then the flow direction and the flow accumulation are calculated (ESRI, 2016d & 2016e). The polygon shapefile gets converted into a raster (ESRI, 2016b). All new raster datasets are saved in the workplace folder. For the calculation, the flow direction, the flow accumulation and the polygon raster are needed. These three are converted into arrays (ESRI, 2016f).

Applying custom functions to data from input

In this section, the two costume functions are summoned. They are executed and the two tables are exported into the workfolder.

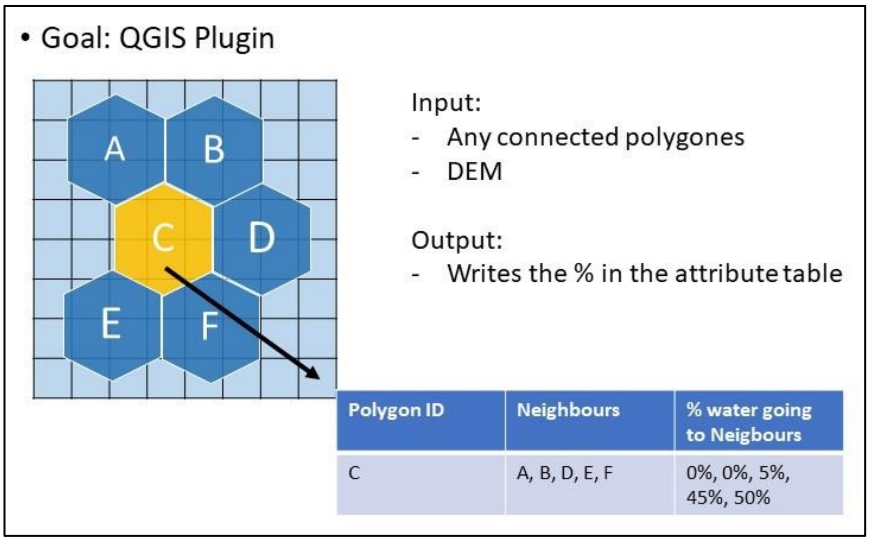
4. Documentation of development process

4.1 Original task

The original task was to program a code which starts with connected polygons of any shape and then to write the percentage of water flowing to each neighbor polygon in the attribute tables of the polygons. The steps for this were thought to be:

1. Identify the neighbors
2. Process the flow accumulation (ex. D-infinity)
3. Process the percentage of water going to the different neighbors
4. Write the percentage to each neighbor in the attribute tables of the polygons

The idea was also, that the code should work with QGIS. To identify the neighbors, the code "Find Neighbor Polygons in a Layer" from Ujaval Gandhi (2019) can be used. Figure 2 shows the original goal graphically.



• Goal: QGIS Plugin

Input:
- Any connected polygones
- DEM

Output:
- Writes the % in the attribute table

Polygon ID	Neighbours	% water going to Neigbours
C	A, B, D, E, F	0%, 0%, 5%, 45%, 50%

Figure 2: Original goal.

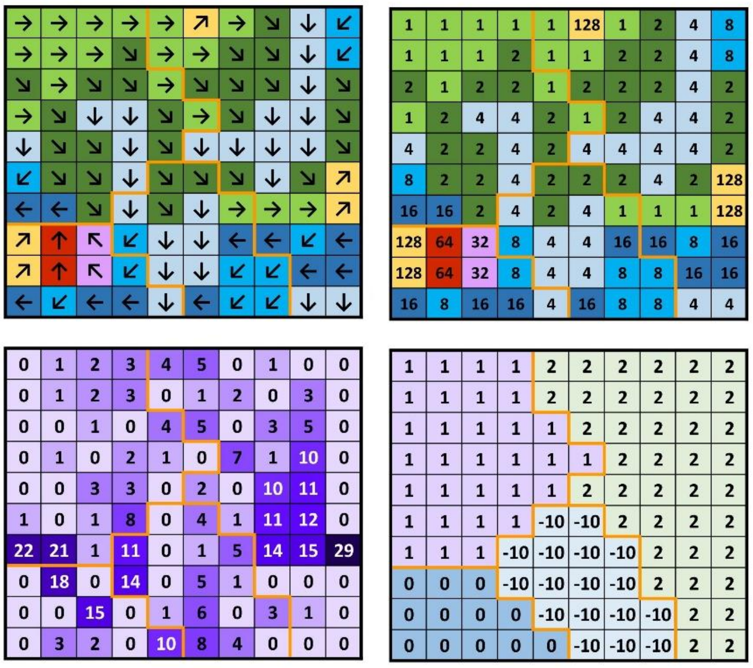
4.2 First ideas

Our first idea was to calculate the flow direction for the whole DEM, but the flow accumulation for each polygon separate. Then we wanted to identify the neighbors and the nodes (boarder between two neighbors). We then realized, that the flow accumulation needs to be calculated for the whole DEM, otherwise a feature like a river is not properly represented. So, we calculated the flow direction and the flow accumulation over the whole DEM. The idea was still to use the nodes to see how much flows from one polygon into the other. To gain the nodes, the tool 'polygon to line' was used. A first problem was here, that not all nodes were recognized as one line, some got divided. The lines were then converted into a raster. At this point, we run into more problems. The obtained, pixelated lines were never continuous. If a buffer around the line was used, the nodes were at some points more than one pixel broad and still had gaps.

We decided to use an entirely different approach. Because it was already late in the semester, we worked thenceforth with ArcGIS instead of QGIS, because we are more experienced with it. The new idea was to create a script tool, which includes the raster calculations as well as a loop which goes through all cells. We first planned to search for every cell in the polygon raster, if one of the neighbor cells belongs to a different raster. If yes, the flowdirection raster is checked. During the first attempt to write this loop, we realized that the code performs way better when we look first in the flow direction raster, in which neighbor cell it flows and then if this cell belongs to another polygon.

4.3 Testing

We decided to test the code first with small arrays. A flow direction array was drawn, and with this, the flow accumulation table was calculated by hand (Figure 3). Then the flow between each of the polygons was determined and written in the connectivity table (Table 2). After the first test run, some numbers in the table were right, but others were too high. The mistake was due to a missing line in the for-loop. We had not defined what happens when a flow direction cell points out of the raster. After we inserted an else statement there to prevent these errors, the code worked as intended and without issues for the small arrays.



→	→	→	→	→	↗	→	↘	↓	↙
→	→	→	↘	→	→	↘	↘	↓	↙
↘	→	↘	↘	→	↘	↘	↘	↓	↘
→	↘	↓	↓	↘	→	↘	↓	↓	↘
↓	↘	↘	↓	↘	↓	↓	↓	↓	↘
↙	↘	↘	↓	↘	↘	↘	↓	↘	↗
←	←	↘	↓	↘	↓	→	→	→	↗
↗	↑	↖	↙	↓	↓	←	←	↙	←
↗	↑	↖	↙	↓	↓	↙	↙	←	←
←	↙	←	←	↓	←	↙	↙	↓	↓

1	1	1	1	1	128	1	2	4	8
1	1	1	2	1	1	2	2	4	8
2	1	2	2	1	2	2	2	4	2
1	2	4	4	2	1	2	4	4	2
4	2	2	4	2	4	4	4	4	2
8	2	2	4	2	2	2	4	2	128
16	16	2	4	2	4	1	1	1	128
128	64	32	8	4	4	16	16	8	16
128	64	32	8	4	4	8	8	16	16
16	8	16	16	4	16	8	8	4	4

0	1	2	3	4	5	0	1	0	0
0	1	2	3	0	1	2	0	3	0
0	0	1	0	4	5	0	3	5	0
0	1	0	2	1	0	7	1	10	0
0	0	3	3	0	2	0	10	11	0
1	0	1	8	0	4	1	11	12	0
22	21	1	11	0	1	5	14	15	29
0	18	0	14	0	5	1	0	0	0
0	0	15	0	1	6	0	3	1	0
0	3	2	0	10	8	4	0	0	0

1	1	1	1	2	2	2	2	2	2
1	1	1	1	2	2	2	2	2	2
1	1	1	1	1	2	2	2	2	2
1	1	1	1	1	1	2	2	2	2
1	1	1	1	1	2	2	2	2	2
1	1	1	1	-10	-10	2	2	2	2
1	1	1	-10	-10	-10	-10	2	2	2
0	0	0	-10	-10	-10	-10	2	2	2
0	0	0	0	-10	-10	-10	-10	2	2
0	0	0	0	0	-10	-10	-10	2	2

Figure 3: Test numpay-arrays. upper-left: flow direction arrows, upper-right: flow direction numbers, lower-right: Flow accumulation, lower-left: polygons.

	0	1	2	-10
0	0	21	0	0
1	0	0	12	14
2	0	0	0	7
-10	26	0	6	0

Table 2: Test connectivity table.

For the second test, we used a snipped of a DEM (2188 x 1500 pixels) of the canton Solothurn (Kanton Solothurn, 2019) and some polygon shapefiles we drew in ArcGIS. Figure 4 shows one set of drawn polygons in front of the flow accumulation raster. With this data, we tested the whole script, including the raster calculations. It took 1.5 minutes to execute the script and the two tables were created without errors.

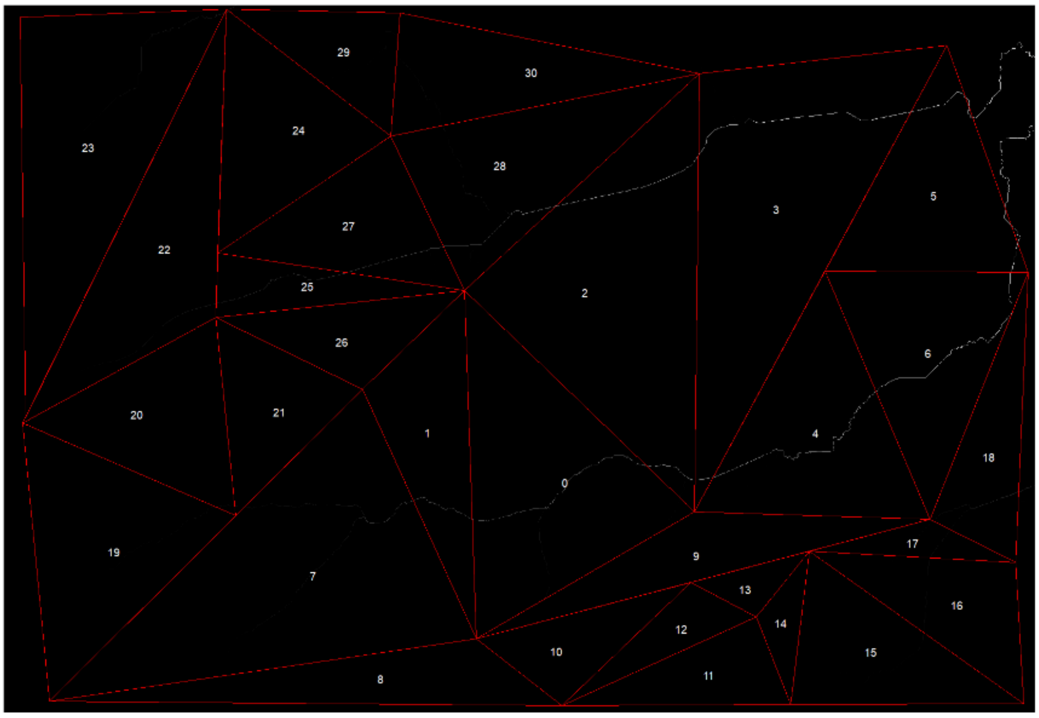


Figure 4: Polygon Shapefile in front of the flow accumulation raster.

References

ESRI (2016a) Creating a toolbox. <http://desktop.arcgis.com/en/arcmap/10.3/analyze/managing-tools-and-toolboxes/creating-a-custom-toolbox.htm>. Accessed at: 02.10.2019.

ESRI (2016b) Feature to raster. <http://desktop.arcgis.com/en/arcmap/10.3/tools/conversion-toolbox/feature-to-raster.htm>. Accessed at: 02.10.2019.

ESRI (2016c) Fill. <http://desktop.arcgis.com/en/arcmap/10.3/tools/spatial-analyst-toolbox/fill.htm>. Accessed at: 02.10.2019.

ESRI (2016d) Flow Accumulation. <http://desktop.arcgis.com/en/arcmap/10.3/tools/spatial-analyst-toolbox/flow-accumulation.htm>. Accessed at: 02.10.2019.

ESRI (2016e) Flow Direction. <http://desktop.arcgis.com/en/arcmap/10.3/tools/spatial-analyst-toolbox/flow-direction.htm>. Accessed at: 02.10.2019.

ESRI (2016f) RasterToNumPyArray. <http://desktop.arcgis.com/en/arcmap/10.3/analyze/arcpy-functions/rastertonumpyarray-function.htm>. Accessed at: 02.10.2019.

Kanton Solothurn (2019) So!GIS Geodaten. <https://geoweb.so.ch/geodaten/index.php>. Accessed at: 02.10.2019.

Ujaval Gandhi (2019) Find Neighbor Polygons in a Layer. <http://www.qgistutorials.com/en/docs/find_neighbor_polygons.html>. Accessed at: 02.10.2019.